

APPLICATION OF NEURAL NETWORK ALGORITHMS FOR BPM LINEARIZATION*

J. Musson[#], C. Seaton, J. Yan, M. Spata, Thomas Jefferson National Accelerator Facility, Newport News, VA, 23606, U.S.A.

Abstract

Stripline BPM sensors contain inherent nonlinearities as a result of field distortions from the pickup elements. Many methods have been devised to facilitate corrections, often employing polynomial fitting. The cost of computation makes real-time correction difficult, especially when integer math is utilized. The application of neural-network technology, particularly the multi-layer perceptron (MLP) algorithm, is proposed as an efficient alternative for electrode linearization. A process of supervised learning is initially used to determine the weighting coefficients, which are subsequently applied to the incoming electrode data. A non-linear layer, known as an “activation layer,” is responsible for the removal of saturation effects. Efficient calculation of the sigmoidal activation function via the CORDIC algorithm is presented as an expedient for implementation of an MLP in an FPGA-based software-defined radio (SDR).

INTRODUCTION

Detection of beam position in accelerators is a mature subject, utilizing a myriad of calculation schemes. RF-based ratiometric methods employing stripline BPMs generally adopt an approximation, since the accelerator beams are designed to remain within a few millimeters of boresight, thus maintaining reasonable linearity. Precision optical models for JLAB's electron beam assume a position accuracy of <100um, placing additional demands on the BPMs. Position calculations are often achieved by difference-over-sum (delta/sigma) or log-ratio approximations, with typical accuracies of 1%.

As part of an ongoing upgrade effort at JLAB, a stripline BPM from ELBE in Rosendorf has been studied as a possible improvement to the standard wire BPMs [1]. The anticipated benefits include reduced cost, improved manufacturability, and an increase of sensitivity from 1dB/mm to 2dB/mm. A prototype stripline was tested as before, and compared with the wireline BPM. A slope of 2dB/mm was observed for sensitivity at boresight, but at the potential cost of nonlinear behavior off-center,

BPM NONLINEARITY

Precise position determination often assumes linearity with respect to the sensor, the electronic detection system, and also the algorithm. However, nonlinearities appear at nearly every point in the chain, which must be considered when attempting to measure higher-order effects on the accelerator beam. A series of recent measurements at JLAB included mapping sextupole fields using BPMs.

The BPM nonlinearities were of the same order of magnitude, and had to be precisely determined, such that they could be removed from the data set [2]. The first step in the process involved assessing the validity of the algorithms, and verifying the empirical constant values. As an example, the delta/sigma and log-ratio methods were employed after data was collected using a Goubau Line test bench, and an Agilent 3-port network analyzer as a detector [3]. The nonlinear aspects were clearly observed, as demonstrated in Figure 1.

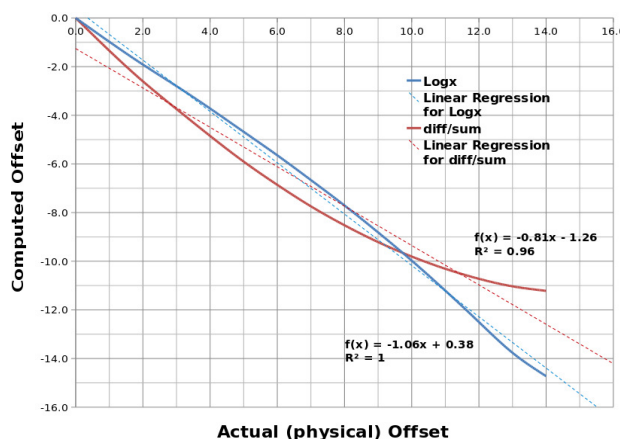


Figure 1: Comparison of BPM position algorithms, demonstrating nonlinear behavior away from boresight.

Stripline BPM

Complete field maps were obtained using the G-Line testbench, confirming the rather large distortions for radial displacements greater than 1 cm within the stripline BPM. Although typical beam steering is contained within the 1 cm² area, the initial concern was that operators would have little information beyond that region, possibly contributing to beam strikes. Although the log-ratio algorithm improved the distortion, the effect still required rectification before insertion into the accelerator. Figure 2 compares resulting field maps, utilizing delta/sigma.

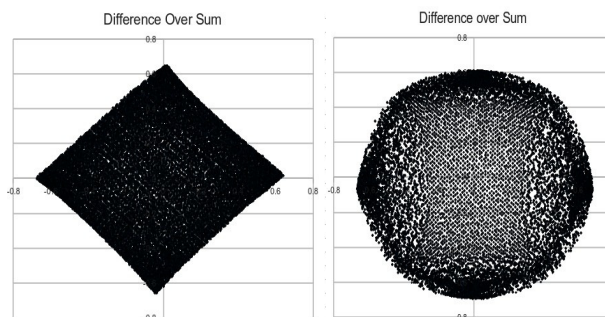


Figure 2: G-Line-derived delta/sigma field maps (both +/-1cm) for wireline and stripline BPMs. Units are in cm.

NEURAL NETWORKS: THE MLP

Although the field of Neural Networks is quite broad, the specific modeling of pattern recognition by Rosenblatt in 1958 resulted in an algorithm known as the Multi-Layer Perceptron (MLP) [4]. Given a set of finite inputs and outputs, the paths, or nets, are interconnected with a hidden layer using weighting coefficients, as shown schematically in Figure 3.

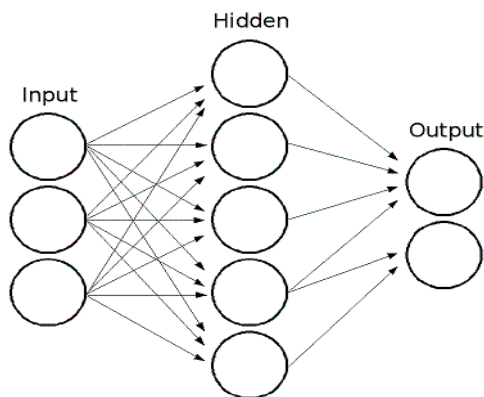


Figure 3: MLP schematic having three inputs, two outputs, and a hidden layer consisting of five neurons.

Although Rosenblatt described the ability of a three-layer MLP to mimic any linear function to arbitrary accuracy, recent utility has been improved with the use of nonlinear logistic sigmoid “activation functions” for the hidden layer, facilitating nonlinear mappings [5]:

$$\varphi(z) = \frac{1}{1 + e^{-\alpha z}}$$

where

z = neuron input

α = saturation weighting

A sigmoid is especially attractive, since it closely mimics the saturation observed by the BPM sensor and associated algorithms, while retaining linear behavior near the origin (ie. BPM boresight). In addition, a sigmoid-based MLP can perform coordinate rotations and trigonometric functions.

MLP Learning

Learning is the process whereby coefficients are adjusted, so as to minimize the cost-function error (eg. RMS) between the current MLP output, and the intended output. Typically, a training-target data set is obtained, either from past experience, or in this case, from a rotary optical encoder employed in the actual field map. Figure 4 compares a slice of the raw field input training data set to the actual encoder target data set. An MLP with four

inputs, a single hidden layer, and two outputs was assigned BPM electrode inputs X+, X-, Y+, Y-, while the outputs were the signed X and Y positions. Although provisions for a third output representing the four-wire sum was available, it was ignored. After an entire measurement, approximately 50 iterations were required to obtain the necessary weightings. A procedure known as output-weight optimization with backpropagation (OWO-BP) simultaneously solves a system of linear equations and updates the coefficients, via a steepest-descent, with each pass [6]. As a calculation expedient, MATLAB was used for the learning process, which employs a similar Levenberg-Marquardt (LM) cost-function method [7].

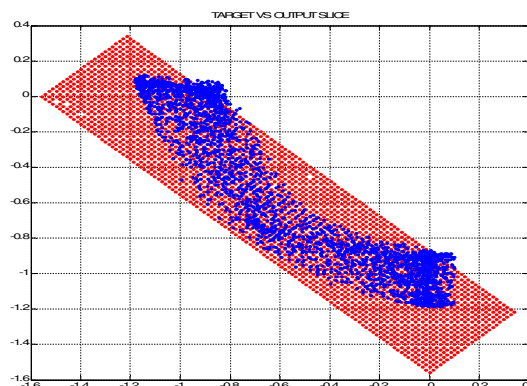


Figure 4. Partial field map of training stripline BPM data, (blue), compared with the target data (red) for supervised MLP learning. Nonlinearity is clearly demonstrated.

RESULTS

The intended result of MLP application was to yield a 1 cm² active region along the BPM boresight. A two-neuron hidden layer was first attempted, minimizing computational overhead. Although the algorithm struggled with the field fringes, a 1 cm² mean-squared error (MSE) of ~500 um resulted, with most of the area <300 um. Although close to the 100 um target, smaller MSE is still required. MSE data is presented in Figure 5.

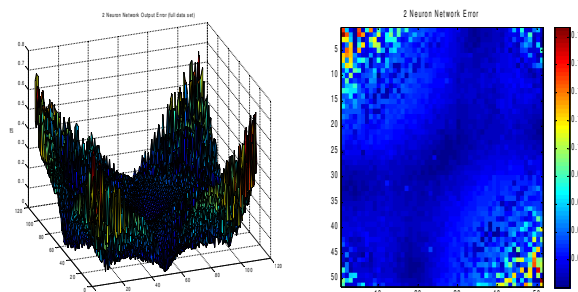


Figure 5. MSE resulting from a 2-neuron hidden layer MLP. The 3-D plot shows blow-up at the BPM corners, while the 2-D plot highlights the ~300 um MSE within the 1cm² active region about boresight. MSE units are in cm.

Increasing the number of hidden layer neurons had the desired effect of reducing the MSE within the 1 cm² active region. Four neurons resulted in a MSE of ~200 um, while eight neurons suppressed the error to ~50 um. In addition, a 45 degree coordinate rotation was easily handled by the eight neuron case, potentially removing the requirement from post-processing. Figure 6 compares the MSE data for four and eight neurons, respectively.

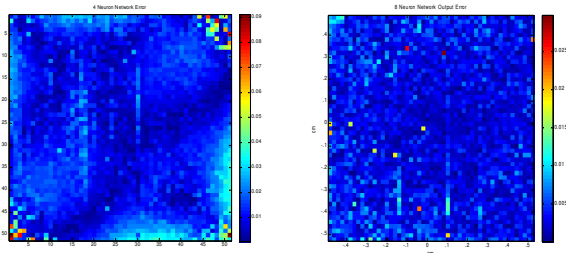


Figure 6. MSE resulting from 4-neuron MLP (left) and 8-neuron MLP (right). MSE largely below 200um is demonstrated for 4-neurons, while 8-neurons suppress MSE to well below 100um. MSE units are in cm.

Over-fitting of the data was slightly apparent at the MLP output map fringes with eight neurons, as evidenced by small whisps resembling solar flares. Single-pixel outliers, as seen in Figure 6, are also possible indications of over-fitting. Further increasing the number of hidden layer neurons resulted in other obvious distortions. It was therefore determined that the least number of neurons resulting in the prescribed MSE provided the best strategy. Computational load is also minimized by performing a minimalization of the resulting MLP.

FPGA Implementation

Unlike large-order polynomial linearization, which generally requires post-processing, the MLP is well suited for use in an embedded software defined radio. Once the MLP weights have been determined via G-line measurements, model and simulation, or even beam-based calibrations, the task of computing the logistic sigmoid functions remains. Fortunately, modern FPGAs contain the necessary arithmetic blocks within their core to simplify the computation by employing approximations or through the use of the ubiquitous CORDIC algorithm [8,9,10]. Since the MLP is also capable of performing rotations, post-processing of rotated or mis-aligned BPMs is potentially avoided.

CONCLUSION

BPM position inaccuracies are a result of sensor nonlinearities, inevitable algorithmic approximations, and analog receiver electronics. BPM sensors are inherently nonlinear, but might otherwise possess attractive attributes. Strategic selection of numerical algorithms,

specifically the MLP, can account for all of these effects, requiring only training and target data sets, followed by a learning procedure. Eight hidden layer neurons resulted in position data sufficient for precision optical models. Subsequent use in a SDR is made possible by the use of available FPGA routines. Real-time processing is therefore realized, eliminating complicated models, curve-fitting, or look-up tables. In addition, the BPM can be trained for specialized applications, possibly involving intentional distortions or additional pre-processing.

ACKNOWLEDGEMENTS

The authors would like to thank the Thomas Jefferson National Accelerator Facility, specifically the Instrumentation and Controls Group, for their generous support of BPM development. In addition, the financial support of the Operations Group is especially appreciated.

REFERENCES

- [1] P. Evtushenko, A. Buchner, H. Buttig, P. Michael, B. Wustmann, K. Jordan, "Stripline BPM Monitors for ELBE," Proc. DIPAC 2001, ESRF, Grenoble, France.
- [2] M. Spata, T. Allison, K. Cole, J. Musson, J. Yan, "Evaluation and Correction of the Nonlinear Distortion of CEBAF Beam Position Monitors," Proc. IPAC 2011, San Sebastian, Spain.
- [3] J. Musson, K. Cole, "Application of *Goubau* Surface Wave Transmission Line for Improved Bench Testing of Diagnostic Beamline Elements," PAC 2009, Vancouver, Canada.
- [4] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, (1958) 65: pp. 386-408.
- [5] C. Bishop, *Pattern Recognition and Machine Learning*, Springer-Verlag, NY, NY., 2006.
- [6] H. Chen, M. Manry, H. Chandrasekaran, "A Neural Network Training Algorithm Utilizing Multiple Sets of Linear Equations," *Neurocomputing*, Vol. 25, No. 1-3, pp. 55-72, April, 1999.
- [7] K. Levenberg, "A Method for the Solution of Certain Non-Linear Problems in Least Squares," *Quarterly of Applied Mathematics* 2: pp. 164-168 1944.
- [8] I. Sahin, I. Koyuncu, "Design and Implementation of Neural Networks Neurons with RadBas, LogSig, and TanSig Activation Functions on FPGA," *Electronics and Engineering* No. 4 (120), 2012. pp. 51-54.
- [9] D. Khodja, A. Kheldoun, L. Refoufi, "Sigmoid Function Approximation for ANN Implementation in FPGA Devices," Proc. of the 9th WSEAS Int. Conf. On Circuits, Systems, Electronics, Control, and Signal Processing, Stevens point, WI. 2010
- [10] D. Anguita, S. Bencetti, A. DeGloria, G. Parodi, D. Ricci, S. Ridella, "FPGA Implementation of High Precision Feedforward Networks," Proc. Of 6th Int. Conf. on Microelectronics for Neural Networks, Evolutionary and Fuzzy Systems, Sept. 24-27, 1997, Dresden, Germany, pp. 240-243.